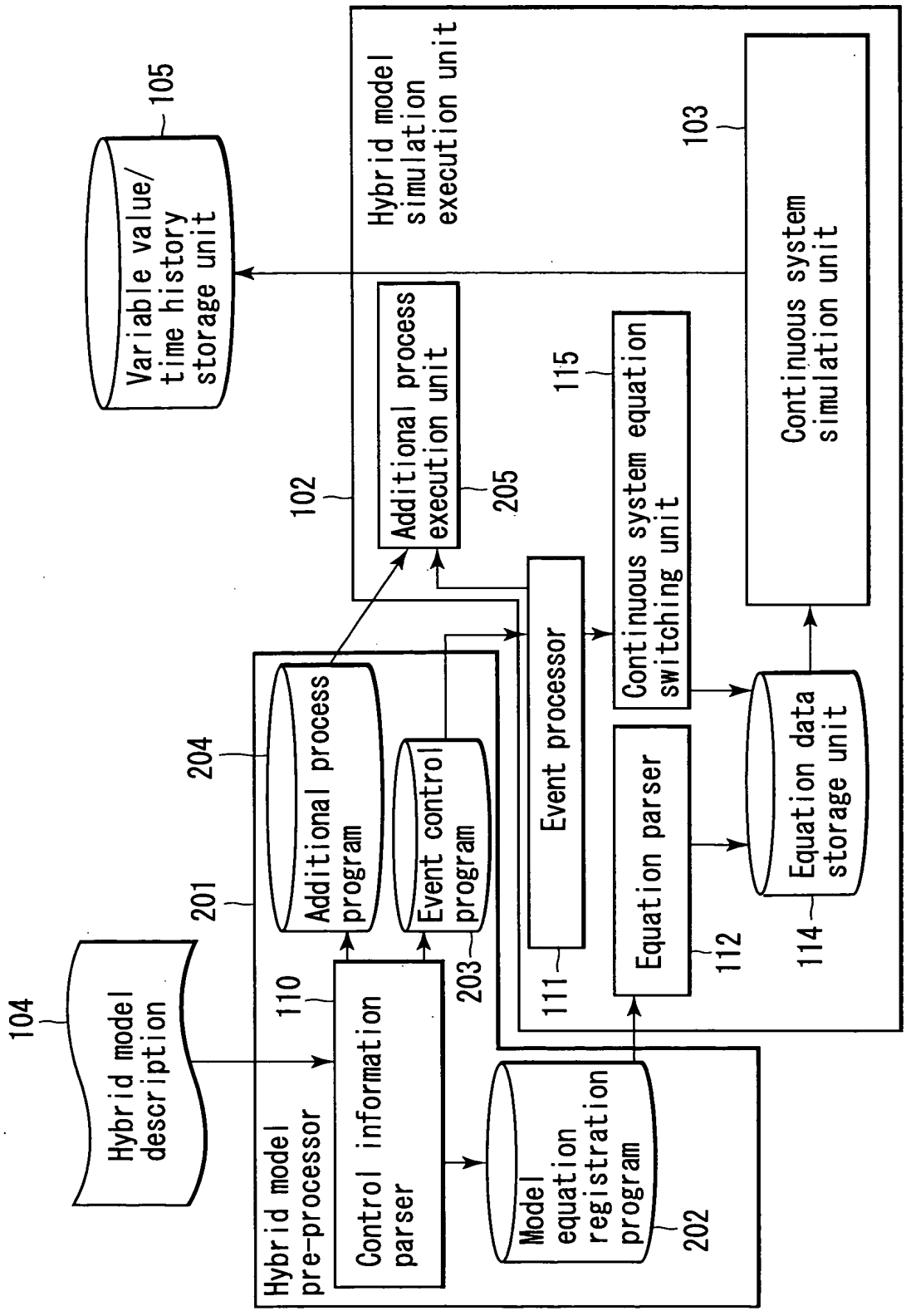

Hybrid model description
104
Hybrid model pre-processor
110
Control information parser
Model equation registration program
202
Additional process program
204
Event control program
203
201
Event processor
111
Equation parser
112
Continuous system equation switching unit
115
Equation data storage unit
114
Additional process execution unit
205
102
Hybrid model simulation execution unit
Continuous system simulation unit
103
Variable value/ time history storage unit
105

FIG. 1

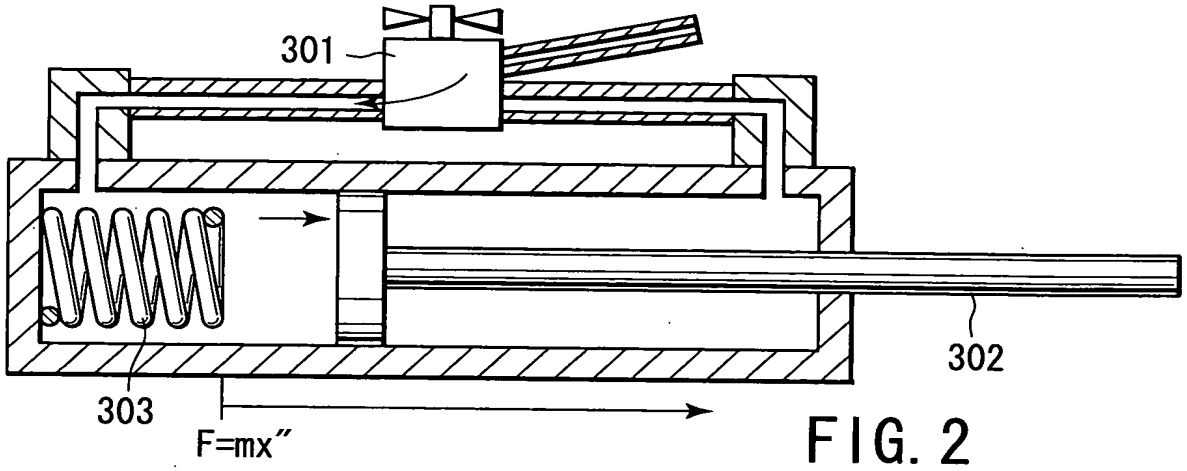


FIG. 2

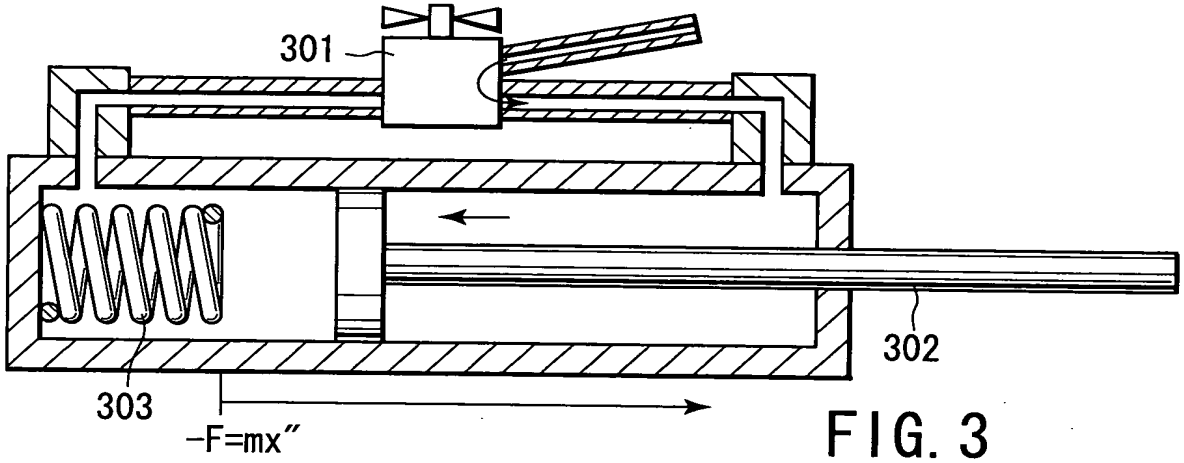


FIG. 3

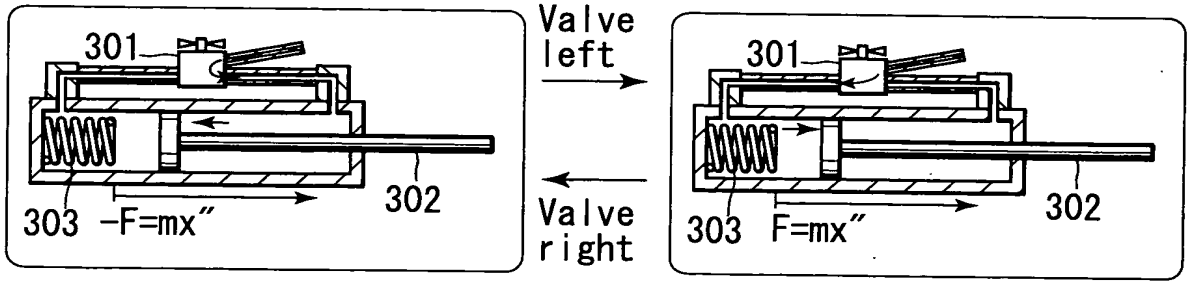
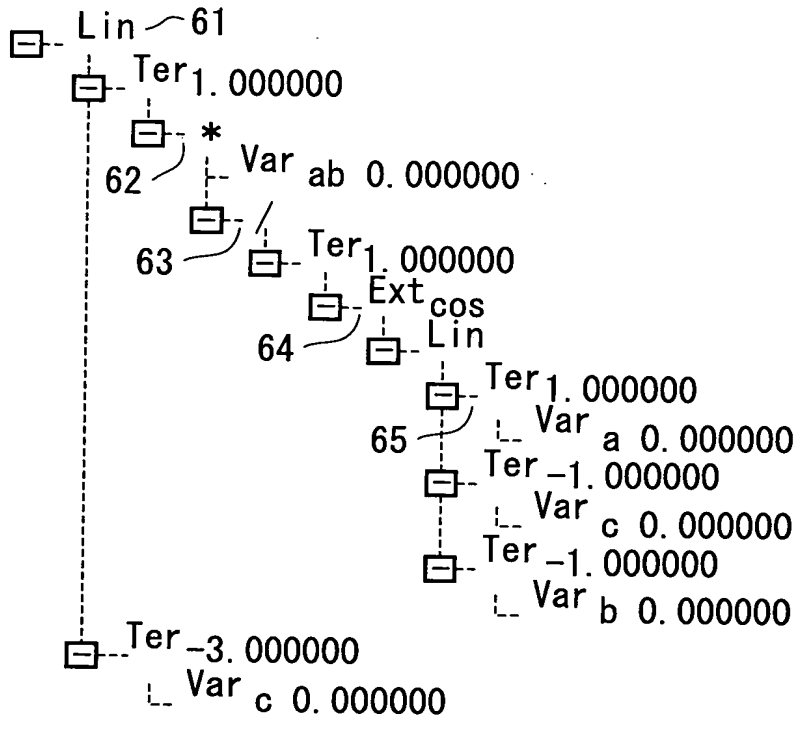


FIG. 4

```
L1   #define m 1
L2   #define f 100
L3   Right, ev1        ev2
L4   wait 50 do Left, ev3                    eq1
L5   always if Left then do always F=m*x" watching Right,
L6   always if Right then do always -F=m*x" watching Left,
L7   sample(x),              ev4
L8   x=0,x'=0,                              eq2
L9   module"C"
L10   {
L11    int cPrint (int num)
L12     {
L13       FILE* fp=fopen("log.txt","a");
L14       fprintf (fp,"%d",num);
L15       fclose(fp);
L16     }
L17    },
L18    int cPrint (int num),
L19    wait 100 do E, ev5
L20    prosess(E){
L21        cPrint(x);
L23    }
```

FIG. 5

FIG. 6


Lin
61
Ter 1.000000
*
62
Var ab 0.000000
63
Ter 1.000000
Ext cos
64
Lin
Ter 1.000000
65
Var a 0.000000
Ter -1.000000
Var c 0.000000
Ter -1.000000
Var b 0.000000
Ter -3.000000
Var c 0.000000

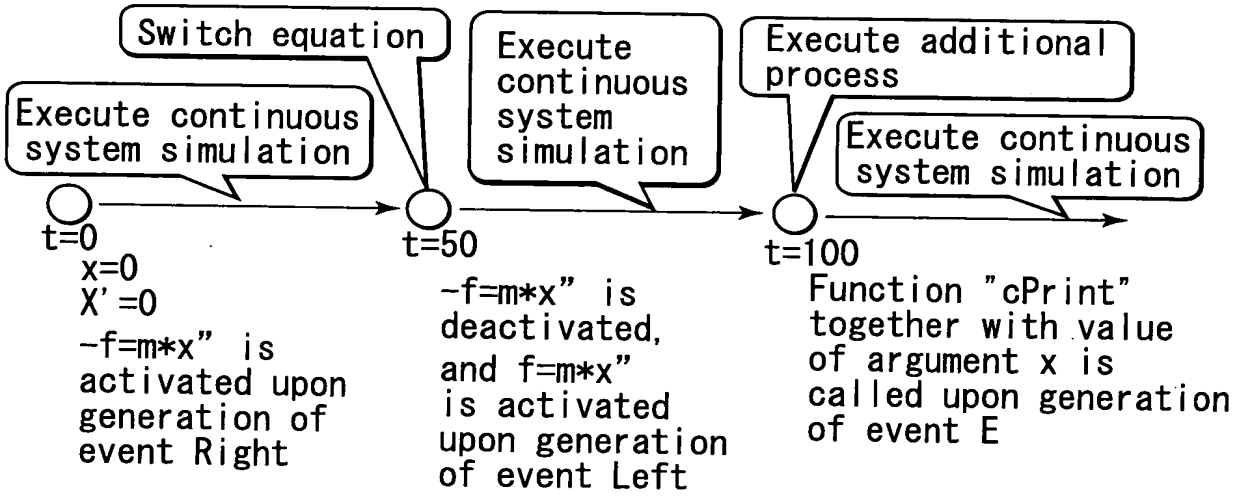

Switch equation
Execute continuous system simulation
Execute continuous system simulation
Execute additional process
Execute continuous system simulation
t=0
x=0
X'=0
-f=m*x" is activated upon generation of event Right
t=50
-f=m*x" is deactivated, and f=m*x" is activated upon generation of event Left
t=100
Function "cPrint" together with value of argument x is called upon generation of event E

FIG. 8

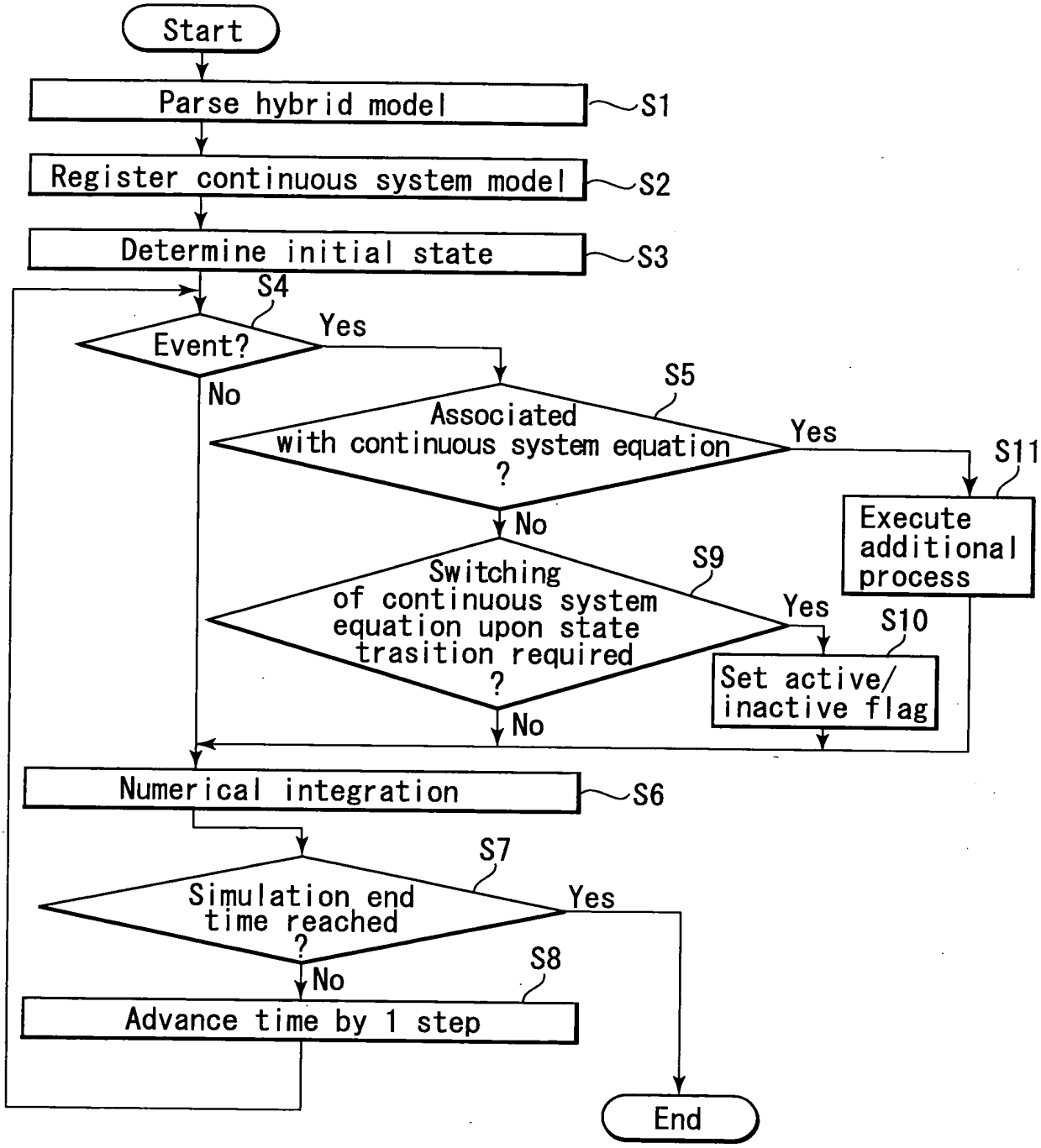

Start
Parse hybrid model
S1
Register continuous system model
S2
Determine initial state
S3
S4
Event?
Yes
No
S5
Associated with continuous system equation ?
Yes
S11
Execute additional process
No
S9
Switching of continuous system equation upon state trasition required ?
Yes
S10
Set active/ inactive flag
No
Numerical integration
S6
S7
Simulation end time reached ?
Yes
No
S8
Advance time by 1 step
End

FIG. 7

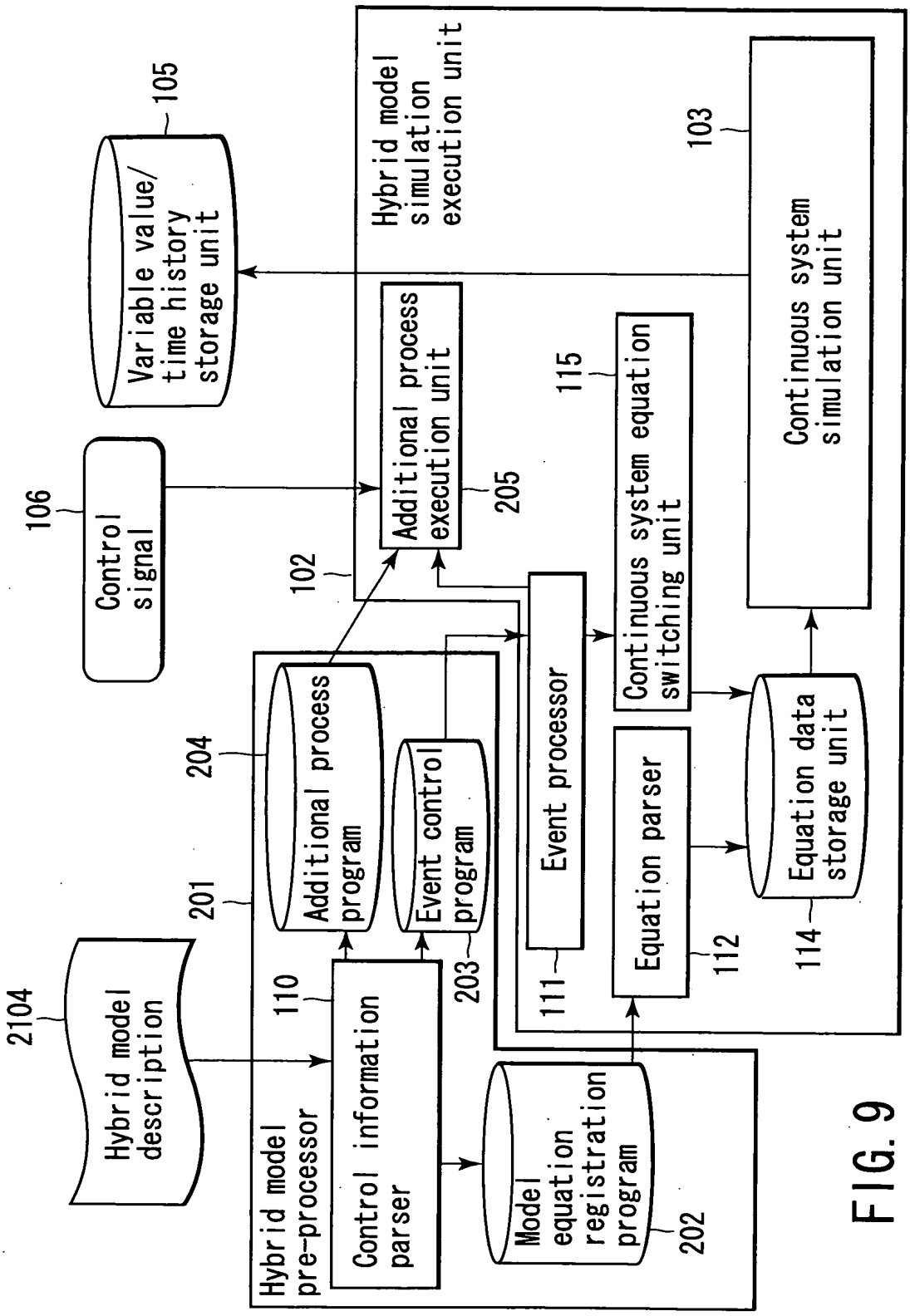

Hybrid model description
2104
Hybrid model pre-processor
110
Control information parser
Model equation registration program
202
201
Additional process program
204
Event control program
203
Event processor
111
Equation parser
112
Equation data storage unit
114
Continuous system equation switching unit
115
Continuous system simulation unit
103
Hybrid model simulation execution unit
102
Additional process execution unit
205
Control signal
106
Variable value/ time history storage unit
105

FIG. 9

```
1   #define m 1
2   #define f 100
3   Right,
4   wait 50 do Left,
5   always if Left then do always F=m*x"  watching Right,
6   always if Right then do always -F=m*x"  watching Left,
7   sample(x),
8   x=0, x'=0,
9   module "C"
10  {
11    int setDataToCtrl( int num,  int data )
12    {
13      outport(num, data);
14    }
15    int getDataFromCtrl( int num )
16    {
17      return inport(num);
18    }
19  },
20  int setDataToCtrl( int num,  int data ),
21  int getDataFromCtrl( int num ),
22  wait 100 do E1,  wait 150 do E2,
23  process( E1 ) {
24    setDataToCtrl( 1,  x )
25  },
26  process( E2 ) {
27    x=getDataFromCtrl( 1 )
28  }
```

FIG. 10

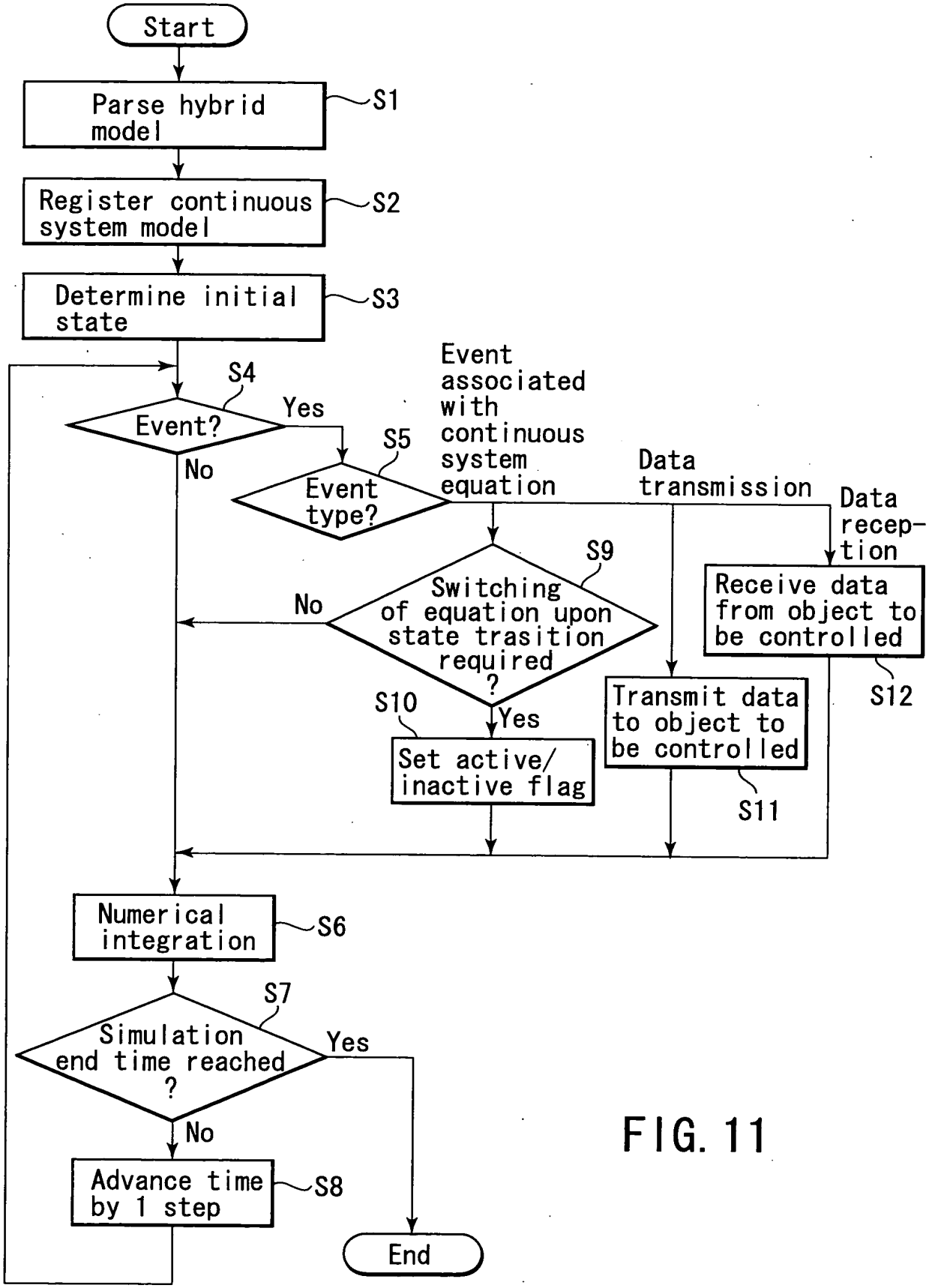

Start
Parse hybrid model
S1
Register continuous system model
S2
Determine initial state
S3
S4
Event?
Yes
No
S5
Event type?
Event associated with continuous system equation
Data transmission
Data recep-tion
S9
Switching of equation upon state trasition required ?
No
Yes
S10
Set active/ inactive flag
Receive data from object to be controlled
S12
Transmit data to object to be controlled
S11
Numerical integration
S6
S7
Simulation end time reached ?
Yes
No
Advance time by 1 step
S8
End

FIG. 11